By Tariz, Co-Founder & ZK researcher at Radius

Radius is building a trustless zk-based shared sequencing layer for rollups. Our biggest advantage is the elimination of harmful MEV, which has been previously introduced in Ethereum research forum.

MEV-resistant ZK-Rollups with Practical VDE (PVDE)[

Layer 2

](/c/layer-2/32)

MEV-resistant ZK-Rollups with Practical VDE (PVDE)@zeroknight , @0xTariz , and @radzin from Radius.xyz Abstract Current MEV solutions based on time-lock puzzles are not viable. Operators cannot easily detect invalid time-lock puzzles or transactions, which can even lead to DoS attacks. We are designing an MEV-resistant ZK-Rollup with a practical approach to MEV minimization using Practical Verifiable Delay Encryption (PVDE). This method generates zk proofs within 5 seconds, a validity proof …

Not only do we protect the funds of rollup users, but we also proposed a model that combines MEV boost with the sequencing layer for rollup revenue generation on the Flashbots forum.

Abstract

The following presents a novel approach to balance the tradeoff between protecting users against harmful MEV (frontrunning, sandwiching) and generating revenue for rollups. The proposed solution is a trustless zk-based shared sequencing layer (developed by Radius) in collaboration with MEV boost to maximize capital efficiency and revenue for rollups while ensuring user transactions remain unaffected.

The proposed solution consists of a two-tiered blockspace structure: Top blockspace is intended for regular user transactions and offers cryptographic protection against harmful MEV, while the Bottom blockspace is designed for builders to carry out revenue-generating activities.

Problem

Rollups face the challenges of balancing the tradeoff between protecting users against harmful MEV and generating revenue from blockspace. While rollups employ a single operator to use a FIFO (first-in, first-out) approach to protect users from harmful MEV, this may sacrifice potential revenue from blockspace and overlook the importance of economically rational actors in contributing to the stability and growth of the rollup ecosystem. Additionally, operators must consistently prove they are adhering to FIFO principles and avoid changing the block order.

Using the bottom blockspace can generate revenue for rollups, but it can also pose a challenge related to user trust. Users must trust that the rollup operator did not exploit the bottom blockspace and subject their transactions to sandwich attacks.

To address these issues, we use a cryptographic method to create a trustworthy environment between users and rollups. This can protect users against harmful MEV in the Top blockspace while utilizing the Bottom blockspace for revenue generation.

Design

We divide the rollup blockspace into two sections:

- Top blockspace: This space is for regular user transactions and provides cryptographic guarantee of harmful MEV protection using PVDE
- Bottom blockspace: Builders can use this space to build the most profitable bundle of transactions by considering the rollup state and transactions in the Top blockspace.

Top Blockspace: Harmful MEV protection

The shared sequencing layer consists of the following components to protect user transactions from harmful MEV:

1. Encryption-as-a-Service (EaaS)

: Enables secure transaction encryption using Practical Verifiable Delay Encryption (PVDE), a zero-knowledge-based scheme that employs time-lock puzzles to generate zk proof on user’s side.

1. Sequencing-as-a-Service (SaaS)

: Sequences encrypted transactions ensuring a fair sequencing mechanism.

The process of protecting user transactions from harmful MEV works as follows:

The user encrypts the transaction and sends it to the sequencer. The sequencer must determine the transaction order and send it to the user before they can decrypt the transaction and view the contents.

User:

1. Creates a transaction
2. Encrypts it using a symmetric key, a solution to the time-lock puzzle
3. Generates zk proof to prove the integrity of the time-lock puzzle and the encrypted transaction with PVDE
4. Sends the time-lock puzzle, encrypted transaction, and zk proof to the sequencer

Sequencer:

1. Verifies the validity of the user's transaction and time-lock puzzle with zk proof
2. Starts solving the time-lock puzzle to find symmetric key (sequentially compute 2^T

)

1. At the same time, the sequencer determines the block transaction order and sends commitment to the user. This must be completed before solving the time-lock puzzle
2. Once the symmetric key is found, the sequencer decrypts the transaction and includes the transaction in the next block

Bottom Blockspace: Revenue generation for rollups with MEV Boost

The sequencer orders the decrypted transactions in the top blockspace and submits them to the mempool. Searchers and builders view the transactions in the top blockspace and use the bottom blockspace to build the most profitable bundles submitting a bid to MEV Boost.

- Searchers

are economically rational actors who specialize in building profitable bundles of transactions using advanced strategies like back-running, arbitrage, and liquidation.

- Builders

create the optimal combination of bundles received from searchers and add them to the bottom blockspace before submitting them to MEV Boost.

- MEV-Boost

selects the block with the highest bid from the blocks received from builders and submits it to the sequencer.

Architecture

[

Architecutre_Radius_sequencing_layer_with_MEVboost

1376×948 81.3 KB

](https://ethresear.ch/uploads/default/original/2X/e/e31c995df9cd80097173c08152a6e92c51e02cc9.jpeg)

Solution Implementation

360° (https://360dex.io) is a DEX that fully implements the structures described earlier, including the core components of the shared sequencing layer (SaaS and EaaS) and the bottom blockspace (batchspace) structure, delivering transactions in batches to validators.

In correspondence to the above structure, the top batchspace contains regular user transactions protected by PVDE, while bottom batchspace with backrunning bundles for revenue generation. The bundles are automatically created using flashloan-based arbitrage strategies without requiring their own liquidity. Transactions in the top batchspace can be opened so that MEV-Boost participants can utilize the bottom batchspace.

Practical Verifiable Delay Encryption (PVDE)

Time-lock puzzle encryption for harmful MEV resistance

Existing time-lock puzzle encryption solutions are impractical due to their high computational costs, which can result in wasted resources and DoS attacks by malicious users if an invalid puzzle is sent. Practical Verifiable Delay Encryption (PVDE) addresses this issue by generating a zk proof for the RSA group-based time-lock puzzle, making it a practical solution for protecting against harmful MEV.

PVDE generates a proof that solving the time-lock puzzle will lead to the correct decryption of valid transactions. This ensures that transaction content is revealed only after the sequencer sequences transactions, delaying the time for the sequencer to find the symmetric key needed to decrypt the transaction and effectively preventing MEV attacks.

The user’s statement is as follows:

πPVDE : The output value (also the symmetric key used for the decryption of the encrypted transaction) is found by computing the time-lock puzzle 2^T

times

The circuit must include the two computations necessary to prove the statement:

1. Time-lock puzzle: $g^{2^T} \ \ mod \ \ N = S_K$
2. Poseidon Encryption: $enc(tx, S_K) \rightarrow C_{tx}$

Details: MEV-resistant ZK-Rollups with Practical VDE (PVDE)

Radius

The Future of MEV-Resistant Cross-rollup Ecosystem

Radius has been part of the Ethereum Foundation Grant developing the core architecture of Practical Verifiable Delay Encryption (PVDE) and successfully demonstrating a PoC for an MEV-resistant DEX.

The shared sequencing layer also enables a latency-free atomic bridge for cross-rollup integrations, fostering further ecosystem growth and stability while remaining resistant to harmful MEV.